DOCUMENT RESUME

ED 347 285 CE 061 420

AUTHOR Kline, Charles R., Jr.; Lester, Mark
TITLE A Model for Integrating a Job-Aiding, Training, and Performance Assessment System--A Preliminary Concept Paper. Final Technical Paper for Period June-August 1985.
INSTITUTION Air Force Human Resources Lab., Brooks AFB, Texas.
REPORT NO AFHRL-TP-87-60
PUB DATE Feb 88
NOTE 80p.
PUB TYPE Reports - Research/Technical (143)

EDRS PRICE MF01/PC04 Plus Postage.
DESCRIPTORS *Computer Oriented Programs; *Evaluation Criteria; *Expert Systems; Job Performance; Learning Modules; *Maintenance; Material Development; *Military Training; Models; *On the Job Training; Postsecondary Education
IDENTIFIERS *Job Aids

ABSTRACT

This paper presents a model for an integrated system used for job-aiding, training, and performance assessment for persons who maintain systems of various types. The model is driven by updatable job aids, by integrated human-machine heuristics, and by an expanding matrix of maintenance activities. The model uses the job-aiding base, updated by computer networks and retrieval systems. In the model, this job-aiding system is part of an expert system. All inputs and outputs are envisioned to be in natural, human languages presented in a user-friendly series of displays and menus. The model also provides for training and performance assessment. To create training modules, the computer subsystem implements the appropriate job aid by presenting it in a training frame. To create a performance assessment battery, the computer subsystem presents the job aid after filtering it through a linguistic transformation that turns it into a case study or, if appropriate, a series of questions. The paper includes 52 references. (Author/KC)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

CE

AFHRL-TP-87-60

AIR FORCE HUMAN RESOURCES LABORATORY

A MODEL FOR INTEGRATING A JOB-AIDING, TRAINING, AND PERFORMANCE ASSESSMENT SYSTEM--A PRELIMINARY CONCEPT PAPER

Charles R. Kline, Jr.
Tracor, M/S 6-8
Tracor Aerospace
6500 Tracor Lane
Austin, Texas 78725

Mark Lester
Department of English
Eastern Washington University
Cheney, Washington 99004

TRAINING SYSTEMS DIVISION
Brooks Air Force Base, Texas 78235-5601

February 1988
Final Technical Paper for Period June - August 1985

Approved for public release; distribution is unlimited.

ED347285

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)
☑ This document has been reproduced as received from the person or organization originating it.
☐ Minor changes have been made to improve reproduction quality.
• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

AIR FORCE SYSTEMS COMMAND
BROOKS AIR FORCE BASE, TEXAS 78235-5601

CE 061 423

BEST COPY AVAILABLE

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely Government-related procurement, the United States Government incurs no responsibility or any obligation whatsoever. The fact that the Government may have formulated or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication, or otherwise in any manner construed, as licensing the holder, or any other person or corporation; or as conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

The Public Affairs Office has reviewed this paper, and it is releasable to the National Technical Information Service, where it will be available to the general public, including foreign nationals.

This paper has been reviewed and is approved for publication.

HENDRICK W. RUCK, Technical Advisor
Training Systems Division

GENE A. BERRY, Colonel, USAF
Chief, Training Systems Division

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) AFHRL-TP-87-60	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Training Systems Division 6b. OFFICE SYMBOL (*If applicable*) AFHRL/IDI	7a. NAME OF MONITORING ORGANIZATION
6c. ADDRESS (*City, State, and ZIP Code*) Air Force Human Resources Laboratory Brooks Air Force Base, Texas 78235-5601	7b. ADDRESS (*City, State, and ZIP Code*)
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Air Force Human Resources Laboratory 8b. OFFICE SYMBOL (*If applicable*) HQ AFHRL	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (*City, State, and ZIP Code*) Brooks Air Force Base, Texas 78235-5601	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. 62205F; PROJECT NO. ILIR; TASK NO 20; WORK UNIT ACCESSION NO 14

11. TITLE (*Include Security Classification*)
A Model for Integrating a Job-Aiding, Training, and Performance Assessment System--A Preliminary Concept Paper

12. PERSONAL AUTHOR(S)
Kline, C.R., Jr.; Lester, M.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (*Year, Month, Day*)	15. PAGE COUNT
Final	FROM Jun 85 TO Aug 85	February 1988	70

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (*Continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUB-GROUP	artificial intelligence; heuristic problem solving; computer network; integrated diagnostics; expert system; job-aiding (Continued)
05	09		
05	08		

19. ABSTRACT (*Continue on reverse if necessary and identify by block number*)

A model for an integrated system used for job-aiding, training, and performance assessment is presented. The model is driven by updatable job aids, by integrated man-machine heuristics, and by an expanding matrix of maintenance activities. The model uses the job-aiding base, updated by computer networks and retrieval systems. In the model, this job-aiding system is part of an expert system. All inputs and outputs are envisioned to be in natural, human, languages presented in a user-friendly series of displays and menus. The model also provides for training and performance assessment. To create training modules, the computer subsystem implements the appropriate job aid by presenting it in a training frame. To create a performance assessment battery, the computer subsystem presents the job aid after filtering it through a linguistic transformation which turns it into a case study or, if appropriate, a series of questions.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL Nancy J. Allin, Chief, STINFO Office	22b. TELEPHONE (*Include Area Code*) (512) 536-3877	22c. OFFICE SYMBOL AFHRL/TSR

DD Form 1473, JUN 86 *Previous editions are obsolete.*

SECURITY CLASSIFICATION OF THIS PAGE
Unclassified

Item 18 (Concluded):

maintenance
man-machine interaction
performance assessment
training

//

AFHRL Technical Paper 87-60 February 1988

A MODEL FOR INTEGRATING A JOB-AIDING, TRAINING, AND PERFORMANCE ASSESSMENT SYSTEM--A PRELIMINARY CONCEPT PAPER

Charles R. Kline, Jr.

Tracor, M/S 6-8
Tracor Aerospace
6500 Tracor Lane
Austin, Texas 78725

Mark Lester

Department of English
Eastern Washington University
Cheney, Washington 99004

TRAINING SYSTEMS DIVISION
Brooks Air Force Base, Texas 78235-5601

Reviewed and submitted for publication by

Hugh L. Burns, Lt Col, USAF
Chief, Intelligence Systems Branch

This publication is primarily a working paper. It is published solely to document work performed.

SUMMARY

In this paper, we present a model for an integrated system used for job-aiding, training, and performance assessment. The model is driven by updatable job aids, by integrated man-machine heuristics, and by an expanding matrix of maintenance activities. Responsible to AFHRL initiatives, as we understand them, and compatible with pending and probable system innovations, the model is designed to serve well into the 21st Century.

The model uses the job-aiding system as the base which is kept up-to-date by computer networked storage and retrieval. In our model, this system is part of an "expert system"; that is, a system which can "learn." All inputs and outputs are in natural, human language presented in a user-friendly series of displays and menus.

The model also provides for training and performance assessment. To create training modules, the computer subsystem implements the appropriate job aid by presenting it in a training frame; to create a performance assessment battery, the computer subsystem presents the job aid after filtering it through a linguistic transformation which turns it into a case study or, if appropriate, a series of questions.

PREFACE

During the Summer of 1984, under the sponsorship of the Air Force Human Resources Laboratory, we worked at the Training Systems Division at Lowry Air Force Base, Colorado. We attempted to create a concretely focused model, based on current knowledge about computer-assisted instruction (CAI), about artifical intelligence (AI), about language and thought. This document and the briefing we gave present the gist of our idea.

Our work was informed by the efforts and support of Dr. Joseph Yasutake, Maj. Dale Baxter, Maj. Richard Bolz, Dr. Roger Pennell, Mr. Brian Dallman and others too numerous to mention. Especial acknowledgement of Maj. Hugh Burns, our host, is conveyed both by this notice and by the document itself--he was the prime force for our coming together; he is a ready listener, an astute critic, and a scholar.

The opinions advanced within this report do not necessarily reflect those of the Air Force, Air Force Systems Command, or the Air Force Human Resources Laboratory (Training Systems Division); they are ours alone.

CRK Jr
ML

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. THE INTEGRATED MODEL	4
A. Job Aiding	4
Classic Expert Systems	4
Interactive Expert Systems	10
Interactive Expert System as a Job Performance Aid	21
A Case Study	23
The Detective System	28
The Interrelationship of Epistemology and Heuristic	30
B. Training	38
How the Training Subsystem Works	40
C. Performance Assessment	43
Interrelationship of Teaching and Testing	43
III. UPDATING THE SYSTEM	46
IV. ANTICIPATING CLEARLY THE FUTURE	49
A. Present - 1995	49
B. Beyond - 1995	51
V. CONCLUSION	54
REFERENCES	55
SELECTED BIBLIOGRAPHY	56

LIST OF FIGURES

Figure		Page
1	System Maintenance Planning	2
2	Model of Expert Systems	5
3	Branching Diagram	7
4	Matching Patient's ISS with Symptom Set	13
5	List of Potential Disease/Ailment Correlation	15
6	Knowledge Matrix	18
7	Interactive Problem Solving	20
8	CRT Display of Initial Symptom Set	24
9	CRT Display of Test Procedures	26
10	Test Procedure Gloss	27
11	Frequency Matrix Drives Probability of "Fix"	32
12	Organizational Support Matrix	33
13	Probe Narrowed to Single Level of Cells	36
14	Interrogative Routine Provides Prompts and Queries in the Detection Mode	37
15	Four Features of Training	39
16	Case Study Arrangement	42
17	Matrix Trace of Sample SRU R&R	47
18	Maintenance on LRU (by ID No.)	48

//

I. INTRODUCTION

As newer systems of all types are brought into use in the Air Force, the job of maintaining them becomes more and more complex. This increased complexity is due both to the increased complexity of the equipment itself and to the necessarily enlarged data base. (New systems do not usually instantly supplant their predecessor systems.)

Maintenance involves a four-fold planning effort: assessing who needs to learn to do what, teaching that job skill, aiding with an up-to-date and well-focused system, and providing the equipment, material, and supervision necessary to allow maintenance personnel to utilize their skills. The first three are training issues as shown in Figure 1. The fourth is an organizational support issue.

In this technical paper, we present a rationale and a concept for dealing with the three training issues; we also survey their impact on those organizational support issues of which we are aware; and we provide examples of how such an integrated system might work.

Crucial to the proposed solution is the use of job aids, computer networked storage and retrieval, and constantly updating and learning systems (so-called "expert" systems). All inputs and outputs (I/Os) of the proposed system are based on use of natural, human language (could be any specific language). We propose a continually updating and learning system with the ability to deal with unprogrammed problem-solutions.

Nothing we have proposed in this document, we believe, relies on technology not in existence today. Indeed, the strength of this model lies in large part in reliance upon state-of-the-art capabilities combined with a unique design for packaging, driving, and using the system.

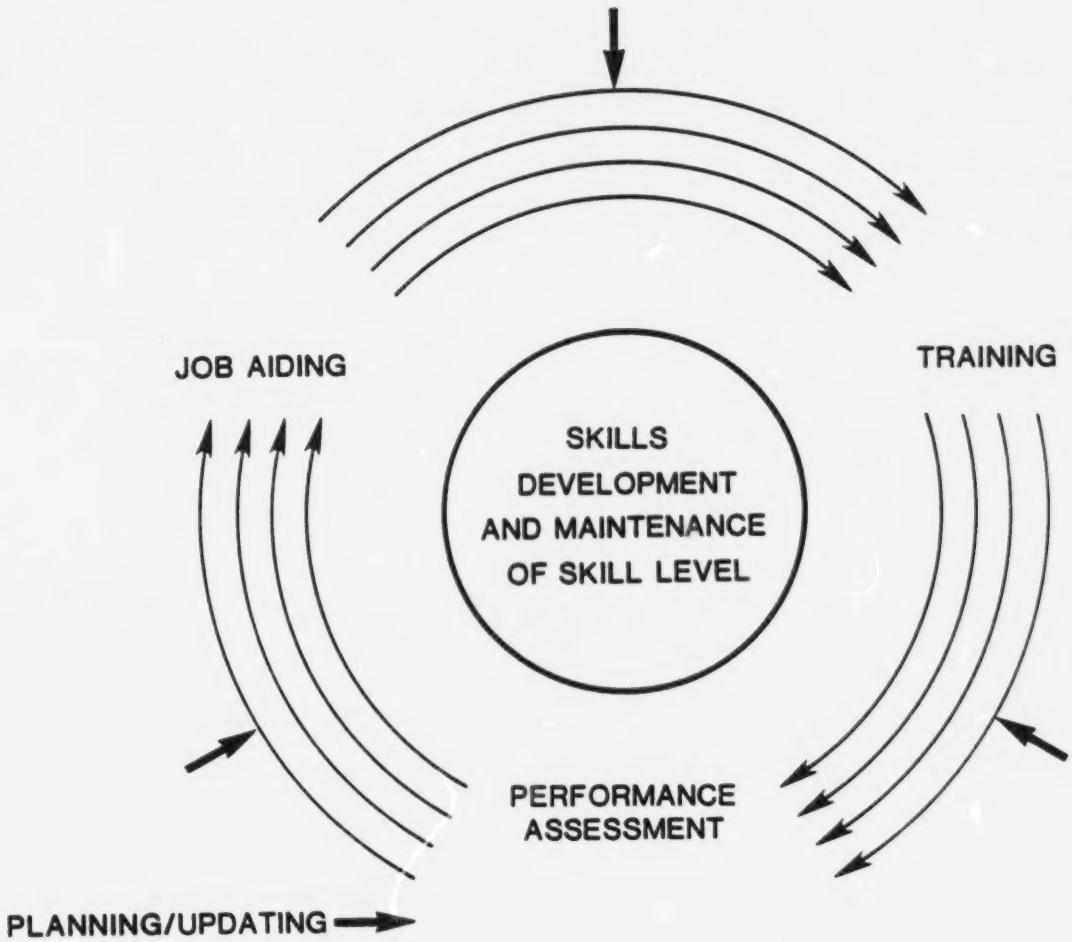


FIGURE 1 SYSTEM MAINTENANCE PLANNING

The redundancy of training and job aiding systems in the Air Force, while certainly understandable, seems to us wasteful and lamentable. One goal of this effort was to eliminate that redundancy by using job aids as the stockpile from which training materials are taken. Training exercises are therefore completely up to date, because the design provides for a continuously updating job-aiding system. These job aids and training exercises form the basis for the non-sociometric aspects of performance assessment. Thus, it is an integrated model. The organizational support requirements are largely data-based; we include a discussion of how that support mechanism might work.

It needs to be stressed that we did not design the actual system. In fact, in our short consulting assignment, we did not study all possible applications of job aids, or of training systems, or of performance assessment. This is a concept paper.

What we have done is work out a solution to this problem: Create the concept for an interactive, computationally and natural language driven, "expert" system which will improve the delivery of training.

We extended the problem to include job aiding and performance assessment and narrowed it to currently available technology (with, to be sure, an eye toward the innovations foreseen for 1985 to 2025).

Our report is presented in three major subsections:

II. The Integrating Model
III. Updating the System
IV. Anticipating Clearly the Future

II. THE INTEGRATED MODEL

The heart of the proposed design is the job aid data base which is called on and augmented for training purposes. The job aid data base becomes also the base upon which performance measurement tools are built. In this section we present the concept of the integrating/integrated model; the presentation is divided into three sections:

A. Job-Aiding
B. Training
C. Performance Assessment

A. Job Aiding

Classic Expert Systems

One of the uses of artificial intelligence (AI) has been to develop expert systems. An expert system is the formalization of the practices (both conscious and intuitive) of experts. These formalizations can then be reduced to step-by-step algorithmic procedures for the guidance of novices, who, with the assistance of the procedures, can then mimic the performance of experts.

Such an approach has certain strengths and weaknesses, as Figure 2 illustrates. Figure 2 shows the interaction of two different dimensions: the skill level of the user of the expert system and the difficulty of the task to be performed. In Cell 1, we see the most powerful use of expert system procedures. The user has a low skill level and thus needs the maximum amount of guidance. The task is highly routine and thus well described and documented by the procedures. As the user gains experience with routine problems, his/her dependence on step-by-step procedure lessens. In Cell 2, we see that a highly skilled individual no longer needs the system at all, or more accurately, his/her requirement for the system changes from needing a guide book to needing a reference book. Thus, an

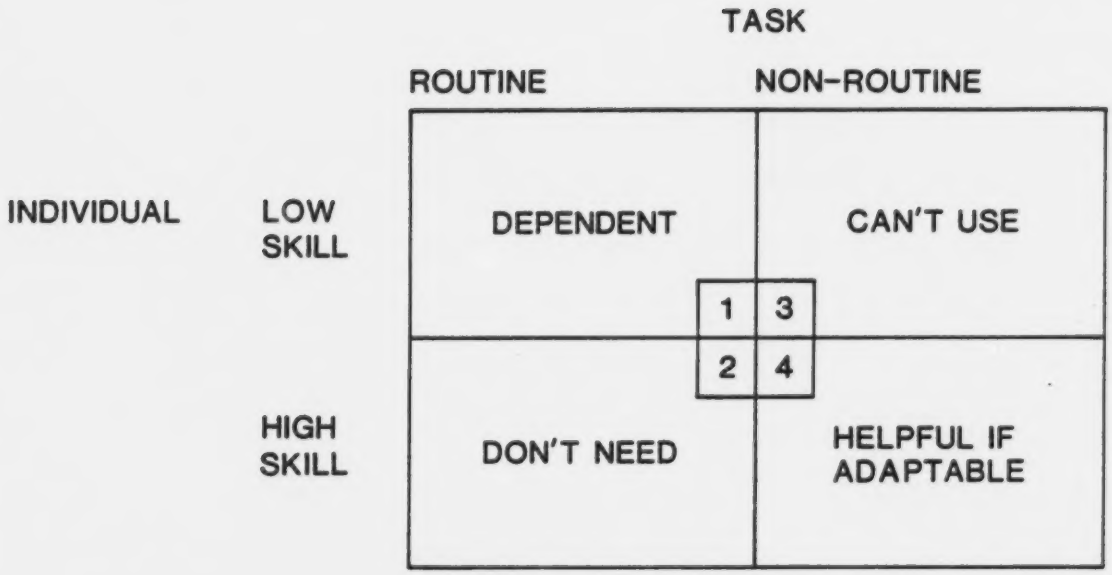


FIGURE 2 MODEL OF EXPERT SYSTEMS

expert system that has a skip-ahead or random access format could serve the needs of the highly skilled individual whereas a highly linear expert system which requires lock-step use would not.

The other dimension in Figure 2 reflects the "fit" of the expert system to the problem or task with which the individual is working. Presumably, the expert system is highly effective in dealing with routine problems. However, not all problems are routine and there will be situations in which the expert system does not "fit" the problem very well. In Cell 3, a novice user cannot use the expert system because the individual lacks the skills to adapt or adjust the system to fit the unique problem. In Cell 4, however, we see that a highly skilled individual is able to deviate from or make adjustments to the expert system to cope with the idiosyncratic nature of the problem at hand. As in Cell 2, the highly skilled individual would be assisted by an expert system that allows for flexible use.

The approach we have described above might be called "the cookbook approach" to expert systems. A novice cook will follow the procedures rather slavishly (Cell 1 in the chart above). As the cook makes the same item over and over, she/he becomes less and less dependent on the recipe, until a point (Cell 2) is reached, at which point the cook merely refers to the recipe to check on a specific ingredient or step in the process. When attempting to prepare an exotic dish for which no reliable recipe exists, the novice is in Cell 3 -- he is unable to adapt existing recipes to meet his needs, whereas a highly skilled chef can combine existing recipes in an innovative way to prepare the dish (Cell 4).

One of the functions of an expert system is to replicate the expert's ability to solve problems. The expert system's approach to problem solving is usually represented as a branching diagram. The top node (or state) in the branching diagram represents a statement of the problem. Branching from this node are two or more nodes which make mutually exclusive statements about the higher node, seen in Figure 3.

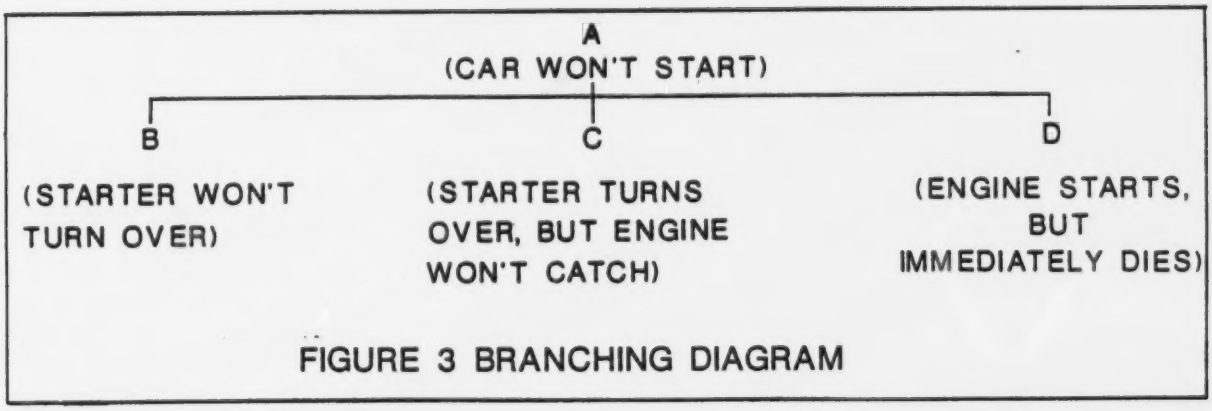


FIGURE 3 BRANCHING DIAGRAM

Subsequent branches will then describe each of the three states (nodes B, C, D) with more detailed statements until an exhaustive set of terminal states, or solutions, are reached. For example, in Figure 3, one terminal state for node B (starter won't turn over) will be the determination that the battery is dead.

In abstract terms, the expert system approach to problem solving may be characterized in the following manner:

1. The expert's knowledge is captured in a branching diagram that has a single initial state (the statement of the problem), a finite number of intermediate states (analyses of the problem), and a finite number of terminal states (solutions to the problem).

2. Branching from the initial state (and from all subsequent states, which are starting points for their dependent branches) are a finite number of mutually exclusive and exhaustive statements about the higher state (or node), only one of which in any given situation can be true.

3. The branching diagram has the technical characteristics of a context-free, phrase-structure grammar (e.g., all intermediate nodes must have branches and each of these branches must end in a terminal state, i.e., a solution; branches must never cross over each other). These characteristics guarantee that from any terminal state there is only one possible pathway back up to the initial state and only one possible pathway from the initial state to a given terminal state (Chomsky, 1963).

Suppose that for some reason the intermediate states (nodes) in Figure 3 were not available to the user of the expert system. All that the user has is the initial state (the problem) and the terminal states (the solution). Could the user still solve the problem? The answer, of course, is "yes," but only in a very inefficient way. The user could find the solution by merely testing each of the terminal states one by one until he

or she found the one that was true. What we have just described is a trial and error system, a system that does not depend on analyzing the problem. Users have no basis to make a connection between the problem and the solution because they lack the intermediate states which are the analyses of the problem.

The expert system we have described rests on two key assumptions: (1) The branches from each node are, in fact, mutually exclusive and exhaustive. For example, in Figure 3, if B is a true statement about A, then C and D cannot also be true statements. Moreover, there is no possibility that there is a fourth statement about A, a new node E, which is also true. Put in different terms, the expert system requires that the sum of the probabilities for all the branches from each node total 100% (i.e., unity). (2) The user of the expert system (the branching diagram) always has sufficient information available to correctly determine which branch is a true statement for the particular problem.

In the real world, neither of these assumptions is valid. Branches do not have to be mutually exclusive -- there can be more than one true statement about a node. For example, in the sample problem, it could be the case that in addition to the battery being dead, there is something wrong with the starting motor. In the real world, it is seldom the case that the sum of all the probabilities for describing a given state is 100%. There is always the chance of discovering a new possibility that the expert system had not anticipated. Finally, we live in an uncertain world. No test equipment is 100% reliable under all conditions. Some symptoms of a problem are intermittent: They appear and disappear. The evidence that one draws upon to make a decision between alternative explanations is often inconsistent and even contradictory.

In the real world, a problem does not have to have a single cause. There can be multiple failures in the same system, and failures can interact with one another with consequences that are quite bewildering to the

observer. The problem and the symptoms of the problem can be disassociated; for example, the failure of a component in system A may be apparent only in a malfunction of a component in system B. In other words, it is not always obvious in which branching tree diagram the problem resides. There is not necessarily a one-to-one correspondence between a problem and its symptoms. For example, the same problem in different circumstances may exhibit different symptoms; conversely, two different problems may have identical symptoms.

The result of this real world complexity and uncertainty is to effectively remove the intermediate nodes from the expert system. In short, real world complexity and uncertainty reduce the expert system problem solving tree to, at best, a trial and error system.

Interactive Expert Systems

The power of the expert system model is that it guarantees the solution to a problem. The limitation of the model is that it requires a set of inputs (a guarantee that the branches from each node are mutually exclusive and exhaustive and that the user always has sufficient information to correctly choose among the branches) that cannot be achieved in the real world in any but the most simplistic circumstances.

We propose a new model of expert system that accepts the limitations noted above. This system does not presuppose that the branches from each node are mutually exclusive and exhaustive, nor does it not presuppose that the user always has sufficient information to correctly choose among the branches. This model assumes that the normal state of affairs in the real world is uncertainty. The user can never assume that the expert system knows all the answers (and even if it did, the user can never assume that he/she has all the information required to find that right answer).

The proposed model is much more realistic in its assumptions about the real world than is the conventional model of expert systems. The model gains in practicality, but loses in power: It cannot guarantee a correct solution to a problem. It provides the user with a set of inputs which the user employs in an interactive manner to attack the problem. It is the user who solves the problem, not the system. The user interacts with the expert system, using the expert system to provide him/her with a process for identifying alternative solutions to the particular problem and a rich body of data which enables him/her to make intelligent guesses about the most likely solution. For this reason, we call the model an "interactive expert system."

The interactive expert system assists the user in moving by stages of successive approximation from initial recognition and definition of the problem to discovery of the solution. The system provides an analytical procedure and a data set which the user can draw upon to find which alternative among many has the best fit with the actual problem. The user must then verify that the best guess is indeed correct. If it is not, the user must then choose the next best guess.

To see how an interactive expert system approximates what experts actually do in solving problems under conditions of uncertainty, we will examine how a physician diagnoses an illness. Let us imagine that a patient comes to a doctor complaining about fatigue and general malaise. Before attempting any specific diagnosis, the doctor draws upon two additional pieces of information: (1) a determination of the patient's general state of health at the present moment, by means of a simple, standardized set of tests (the patient's vital signs): height, weight, pulse rate, body temperature, and blood pressure; and (2) a determination of the patient's medical history by means of a standardized self-reporting form. At this point, the doctor has three pieces of information: 1) the patient's initial description of symptoms, 2) the patient's current state of health (the vital signs), and 3) the patient's medical history. We will collectively call these three pieces of information the patient's initial symptom set (ISS).

The first step in the physician's diagnostic procedure is to match the ISS against the symptom set of diseases and ailments of which the doctor is aware. A critical component of the doctor's expertise is his/her knowledge of diseases and ailments, together with the symptoms that are associated with each of them. In more formal terms, the doctor does a matching sort, selecting for further consideration those diseases and ailments that have symptoms compatible with the ISS and discarding from consideration those that do not match the ISS. This matching sort accomplishes two things: 1) It reduces the number of potential solutions to the diagnostic problem to a tractable number, and 2) the doctor can draw on knowledge of these potential diseases and ailments to make predictions about additional symptoms that the patient may have beyond the ones already identified in the ISS. For example, imagine a patient with three salient symptoms (A, B, C) -- the ISS -- and six diseases and ailments that also have these same three symptoms. We may represent this knowledge as seen in Figure 4. The doctor can now begin to further narrow the list of potential diseases and ailments by seeing if the patient exhibits the additional symptoms D, E, F, and G.

The ability to create the list of potential diseases and ailments which match the ISS is a critical step in the diagnostic process. Without this list and the additional symptoms associated with the diseases and ailments on the list, the doctor has no systematic basis for further diagnosis except by trial and error. The additional symptoms are generated by this list. Without the list, the doctor would have no further symptoms to investigate. In other words, additional symptoms do not exist until the doctor knows to look for them -- a fact especially true of symptoms that are not physically apparent to the doctor, for example, previous events in the patient's medical history that were not elicited on the standard medical history form.

This point is nicely illustrated by an anecdote in one of James Herriot's stories. Herriot was treating a dog with the ISS of listlessness, loss of appetite, vomiting, and diarrhea. Despite a variety of treatments,

PATIENT'S ISS	LIST OF POTENTIAL DISEASES AND AILMENTS WITH SAME ISS					
	1	2	3	4	5	6
A	A	A	A	A	A	A
B	B	B	B	B	B	B
C	C	C	C	C	C	C
	D	E	D	D		G
			F	E		
ADDITIONAL SYMPTOMS:				F		

FIGURE 4 MATCHING PATIENT'S ISS WITH SYMPTOM SET

the dog was getting worse and worse. One day when Herriot was examining the dog, the dog vomited peculiarly forcefully, which Herriot immediately recognized as a key symptom of an ailment that he had not previously considered. When Herriot asked the dog's owners why they had not told him how the dog had been vomiting, they replied that he had never asked them.

The physician's ability to create the list of potential diseases and ailments is highly significant in two other ways: 1) the doctor has a general sense of the relative probability of occurrence of the diseases and ailments on the list. Some may be common; others, quite rare. All other things being equal, the doctor will make a best guess that the more common items are the cause of the problem. 2) The doctor knows or can easily find laboratory tests which can clinically confirm the existence of the items on the list. All other things being equal, the doctor would first employ the tests that would confirm (or disconfirm) the presence of the higher frequency/most likely items on the list. We may represent the doctor's knowledge of each disease and ailment on the list as shown in Figure 5.

Associated with each disease or ailment is (1) a full set of symptoms, (2) a probability of occurrence as correlated with other variables such as age and sex of the patient, (3) a set of laboratory tests for confirming or disconfirming the presence of the disease or ailment, and finally (4) a set of treatments appropriate for this disease or ailment. All things being equal, the next step in the diagnostic process would be to use the clinical tests to confirm the presence of the disease or ailment that has the highest probability of occurrence.

However, all things are not equal. Both the suspected disease or ailment and the laboratory tests to verify best guesses are substantially affected by other variables -- variables of practical importance to the doctor. There are two variables that affect the probability of a disease or ailment. The first is practicality. Suppose that the diagnosis has been narrowed to a choice between two possible ailments. Suppose further that

DISEASE OR AILMENT	SYMPTOM SET	PROBABILITY CORRELATES	CLINICAL TEST SET	TREATMENT SET
1	A	AGE	1	1
	B	SEX	2	2
	C		3	3
	D		4	

FIGURE 5 LIST OF POTENTIAL DISEASE/AILMENT CORRELATION

15

the treatment of the two ailments is identical. In the real world, it may be a matter of only academic interest which of the two ailments the patient is actually suffering from; the doctor has no practical reason to continue the diagnosis further. For example, the doctor might narrow the list of possible diseases to a number of different bacterial infections. If the treatment for all of these different types of bacterial infections is a broad spectrum antibiotic, it is unnecessary to identify which one it actually is.

The second variable affecting the choice of a disease or ailment is what might be termed the worst case situation. Some diseases and ailments are much worse than others. If a life-threatening disease matched the patient's symptom set, the doctor might well override all considerations of probability in order to reassure himself that the patient was not infected with this particular disease.

There are also at least two important variables in selecting a laboratory test. One is the cost of the test. There are many kinds of costs -- cost in time, cost in money, cost in discomfort to the patient. Some tests are harmful and some even dangerous. Another is the reliability of the test itself. In addition to the possibility of laboratory error, some tests are inherently far from being 100% reliable.

The physician must weigh many variables in test selection. There is no algorithm to tell him the best alternative. What is best in one situation may be a poor choice in another situation. For example, the level of risk of a particular test may be acceptable with a young patient but unacceptable with an elderly patient; the doctor may be reluctant to employ an expensive battery of tests for a patient not covered by a health plan; it might be of no practical consequence to even diagnose the ailments of an elderly patient with a defective immunological system.

The physician's analysis of a patient's illness demonstrates how a complex diagnostic procedure works. The key idea of the interactive expert system model is a matrix that associates the possible causes of the problem with (1) the characteristic symptoms of each possible cause, (2) the probability of the occurrence of each cause relative to other causes, (3) tests by which the existence of each cause can be verified, and (4) a set of treatments which remedy or correct each cause. Such a matrix, a "knowledge matrix," is depicted in Figure 6.

This matrix reflects the collective knowledge of experts in the field. One of the great advantages of organizing information in a matrix format is that each cell can be independently updated to reflect new findings or conditions. Another major advantage of the matrix is that the user can treat each column as an independent variable. As in the case of the doctor's diagnostic process, making a best guess about which is the most likely cause in a particular situation is not simply a matter of probability; the best guess must take into account many practical matters of testing and treatment.

An interactive expert system, so called because it requires the interaction of the knowledge matrix with a user, requires that the user do the following in order for the model to succeed:

1. The user must be able to construct some kind of appropriate ISS for the system under analysis. All ISSs will have the same three basic elements: (a) a set of specific symptoms that indicate that there is a problem (if there were not symptoms of this type then one would never know that a problem existed), (b) some information about the current state of the system, i.e., something that corresponds to the vital signs in medical diagnosis, and (c) some information about the past history of the system. In any particular universe (for example, medical diagnosis or troubleshooting a weapons system), there needs to be some standard format for organizing and presenting the ISS.

CAUSES	1 SYMPTOM SET	2 PROBABILITY SET	3 TEST SET	4 TREATMENT SET
1				
2				
3				
N				

FIGURE 6 KNOWLEDGE MATRIX

2. The user must match the ISS against the Symptom Set (column 1) of the knowledge matrix. In a large and complex system with many possible problem causes, this could be done by a computer search for those causes whose symptom set includes the symptoms in the ISS.

3. The user must further narrow the list of possible causes by looking for previously unnoted symptoms in the system under analysis on the basis of the additional symptoms in the Symptom Set of the knowledge matrix.

4. The user must now make the best guess among the remaining possible causes. While the best guess is strongly influenced by the Probability Set (column 2), the user must also take into consideration the practicalities involved in the Test Set (column 3) and the Treatment Set (column 4).

5. The user must verify whether the best guess was, in fact, correct. The user may employ one or more tests from the Test Set, or if the treatment is inexpensive and simple, the user might skip over the testing state entirely and see if the treatment solves the problem. If the best guess was not correct, the user must go back to step 4 and make a new best guess, and so on until either (a) one of the best guesses about the cause of the problem provides the solution, or (b) all of the causes have been eliminated.

In this latter case, the user enters into an entirely different diagnostic procedure which we have labeled the "detective mode." The flow chart (Figure 7) illustrates the entire process of interactive problem solving using the knowledge matrix. We shall conclude this general discussion of the interactive expert system by comparing the characteristics of a conventional expert system with the interactive model. As we stated previously, the conventional expert system is essentially a branching or decision tree model of problem solving. The formal properties of such a model are well known: It has the characteristics of a context-free,

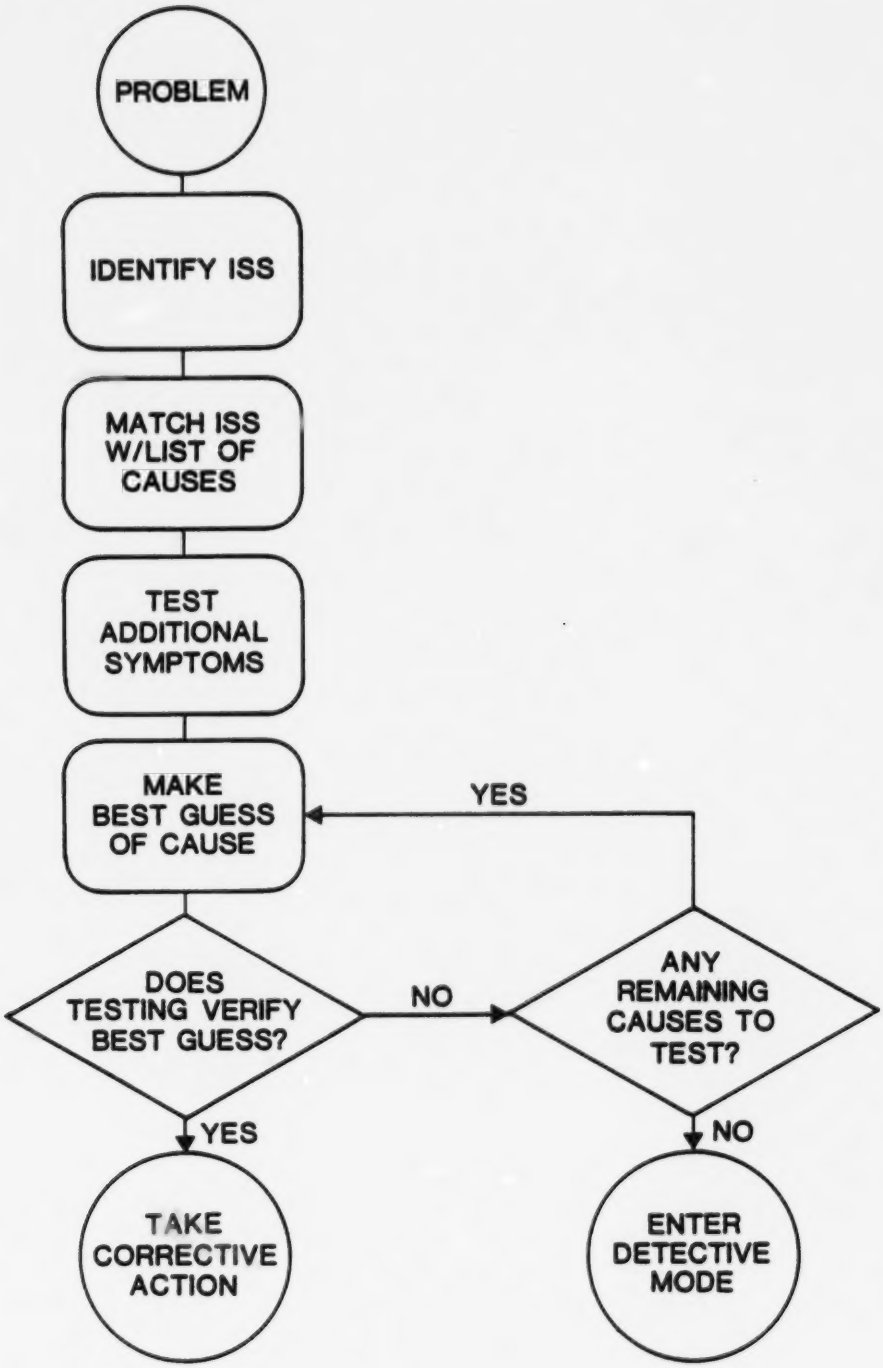


FIGURE 7 INTERACTIVE PROBLEM SOLVING

phrase-structure grammar. One of the most powerful characteristics of a context-free, phrase-structure grammar is that it is deterministic; i.e., there is only one possible solution for any given problem.

On the other hand, the interactive expert system model is a matrix rather than a tree. It is non-deterministic: It identifies a finite number of potential solutions (causes) to the problem and associated with each, a probability of the solution being the correct one. The identification of the correct solution is essentially made by trial and error, albeit an educated trial and error. The interactive model requires much greater contribution from the user than the classical model does. In the classical model, the user's task is to choose among mutually exclusive and exhaustive alternatives at each step in the diagnostic process. In the interactive model, the user must weigh a number of independent (but interactive) variables against each other in the process of making the best guess about what the problem is. In this sense, the classical model of expert systems comprises a deductive process whereas the interactive model comprises an inductive process.

While the classical model is theoretically more powerful (since it guarantees a solution), in practice it reduces to, at best, a trial and error system because the model rests on the assumption that all choices within the decision tree are mutually exclusive, exhaustive, and decidable by the user. The interactive expert system, we believe, is a much more realistic model of how experts actually make decisions under conditions of uncertainty. Moreover, it allows the user to make a number of trade-offs reflecting the conflicting demands and expediencies of real world situations.

Interactive Expert System as a Job Performance Aid

The nature of the knowledge matrix makes the interactive expert system a powerful job performance aid:

1. Since each cell in the knowledge matrix is free standing, the information in the cell can be updated or completely revised without the need to alter the rest of the matrix. Changes in procedures can be fed directly into the computer program that controls the knowledge matrix, allowing for prompt, reliable and relatively inexpensive updating.

2. Each cell can be a window through which a variety of information can be transmitted. For example, a cell for a given Test Set displays several types of tests for that problem. The test names could be used as menus, which, when selected by the user, give additional information about the test: how to conduct it, special equipment needed, time required, where to go to get further information about the test, etc. In a cell for a Treatment Set, the menu could include information about other components to check or adjust as a result of the correction of the original fault -- a critically important point since so many systems interact with other systems.

3. Unlike a classical expert system which sequences its information flow, the knowledge matrix arrays its information in a single display so that the user has immediate access to just those pieces of information that are immediately relevant.

4. As mentioned before, there are a number of trade-offs to consider in the process of diagnosing, testing, and correcting problems. In order to make the best trade-off, the user must have a display of the alternatives and their consequences, a display that is highly compatible with a matrix format but very difficult to represent in a branching diagram. The knowledge matrix allows the user to consider all the options at one time. In a classical expert system, the user is allowed to see trees one at a time, but never the forest.

5. The program should have the capability of recording the results of each use by means of a simple record keeping system. This information could be periodically consolidated at a common site and then used to update the entire system based on the actual live experience of the users.

A Case Study

To complete our discussion of this primary function of the integrated model working in the aiding mode, we present an hypothetical case -- that of the accidental firing of a flare from the yet to be produced ALE-47 Countermeasures Dispenser Set (CMDS) aboard an F-16.

For our presentation, let us assume that an accidental firing of a flare has occurred on the flight line at Bentwaters AB, England. No personnel were hurt, no property destroyed (except one flare), and the maintenance team begins an investigation and plans for repair of the system. (Flares firing while aircraft are on the ground prompts serious reaction -- people on the flight line could be killed; fire can destroy an aircraft and even adjacent aircraft. The team wants to isolate the problem, fix it, and report that it was fixed quickly.)

After entering into the computer the problem (ALE-47 FLARE FIRED, A/C ON GROUND), the tasked personnel are presented with range and environment data queries. These data queries are answered, putting the problem and the surrounding events immediately into the data management system. (This action corresponds to Step One presented in the previous section.)

The CRT then lists the framework for the Initial Symptom Set (ISS) on the left side of the screen and arrays the symptom sets of record which include the ISS as shown in Figure 8.

This display tells the operator that based on only the three symptoms listed by him/her in the ISS, the problem has never occurred. Thus, the operator goes back and checks the Weight-on-Wheels switch (WOW) and the Safety Pin. The WOW is jumpered and the Safety Pin is not inserted. Confirming that two additional symptoms exist (which violates SOP), the operator then reads them into the display. The display then shows that only two faults contained in the organization maintenance system have these

TEST SET 8353 JOB AID USAF DATE: TIME

POSSIBLE KNOWN FAULTS AND RELATED DATA

ISS	FAULT SAF SW	FAULT SEQ SW
1. BIT --- SEQ SW	BIT--- SEQ SW	BIT --- SEQ SW
2. BIT --- SAF SW	BIT--- SAF SW	BIT --- SAF SW
3. POWER ON SYSTEM	POWER ON SYSTEM	POWER ON SYSTEM
	WEIGHT-OFF-WHEELS SW	WEIGHT-OFF-WHEELS SW
	SAF PIN OUT	SAF PIN OUT
		PROGRAMMER CCA FAULT
	P = .05 TEST: 0287	P = .95 TEST: 233

(THIS IS A LIMITED SET OF SAMPLINGS: UNCERTAINTY LEVEL IS .40.)

FIGURE 8 CRT DISPLAY OF INITIAL SYMPTOM SET

symptoms, and that one of them has a 95% probability of being correct. (Since the system is limited by its inputs, it notes at the bottom that, based on normal frequencies and the number in these cells, it predicts the current values above may be wrong as many as 4 times in 10.) Further, it notes what test procedures are used to determine the fault.

Most test procedures will be done at the test bench, and the test procedure is called in two stages, the first of which is the gloss which indicates what Special Test Equipment (STE), time, and other costs may be involved in the test. This allows the operator to decide among different tests when necessary. In our example (see Figure 9), these decisions are not really an issue.

The test with the .95 probability is chosen, and the operator (who has some pressure to know the right answer, and who has time) performs the fault isolation test, finds that the Shop Replaceable Unit (SRU) Programmer Circuit Card Assembly (CCA) is faulty, replaces it with a new one, then retests the Line Replaceable Unit (LRU) and finds that it works.

The AID MODE would have made available a menu of the tests, diagrams, and other references in the library throughout this process--perhaps by a window at the bottom of the screen.

When the operator comes back to the terminal and calls in the case number (probably by maintenance number or by a personal number), the terminal brings up the Test Procedure gloss and begins its data logging queries, illustrated in Figure 10. (We assume in this example that other operators may have logged onto the AID system while the maintenance person was conducting test procedure 0233.) For the largest number of maintenance and troubleshooting activities, this AID system will do well and will provide tremendous service for a large number of installations. Earlier in this paper however, we had a branch on the flow diagram which ended after symptoms and causes could not be found or were exhausted, and that was called DETECTIVE. We discuss it next.

TEST PROCEDURE 0233 SEQ SW FAULT ISOLATION

TIME INVOLVED	COST	SITE	NOTES/POSSIBLE DAMAGE
20 MIN	N/A	N/A	ONLY 3 MIN INVOLVED IN R&R LRU 20 MIN REFERS TO FAULT ISO. R&R OF SRU---PROB. PROG CCA ***INSPECT WOW & SAF SW BEFORE AND AFTER CONDUCTING THIS TEST

FIGURE 9 CRT DISPLAY OF TEST PROCEDURES

TEST PROCEDURE 0233 SEQ SW FAULT ISOLATION

TIME INVOLVED	COSI	SIE	NOTES/POSSIBLE DAMAGE
20 MIN	N/A	N/A	ONLY 3 MIN INVOLVED IN R&R LRU 20 MIN REFERS TO FAULT ISO. R&R OF SRU--PROB. PROG CCA ***INSPECT WOW & SAF SW BEFORE AND AFTER CONDUCTING THIS TEST

TEST COMPLETED: Y N
TIME TAKEN:_____ MIN
TESTS OUT OK? Y N
WHICH CCA? PROG PWR SUPPLY MUX BUS INTERFACE
DOES THIS COMPLETE THE MAINTENANCE ACTIVITY? Y N

NOTE SAFETY REMINDER ABOVE IN THE RIGHT COLUMN!
SYSTEM OFF? Y N

FIGURE 10 TEST PROCEDURE GLOSS

The Detective System

Thus far we have described a model of an interactive expert system which can assist the user in diagnosing faults in complex equipment. The heart of this system was a knowledge matrix which provided the user with the known causes associated with the symptoms the user identified (ISS), along with additional symptoms and tests that the user could use to confirm or disconfirm each potential cause. Based on a probability table and other practical considerations, the user went through the list of possible causes, one by one, using the tests provided until the correct cause was identified and the fault corrected.

In this section, we deal with the situation in which the user has gone through the possible causes listed in the knowledge matrix and has eliminated all of them without solving the problem. In this situation, one of the following three conditions must be true:

1. The solution to the problem is, in fact, in the existing knowledge matrix, but for some reason the user overlooked it. We call this condition "user error."

2. The solution to the problem is in the interactive expert system data base but not in the knowledge matrix that the user called up. We call this "symptom error," because the user did not use the right configuration of symptoms in the ISS.

3. The solution to the problem does not exist in the interactive expert system. That is, even if the user were to go through every cause in the system one by one, the user would still not find the solution to the problem. We call this "system error."

The DETECTIVE system is a procedure for exploring each of these three error conditions in a manner that is the most practical help to the user. The difficulty is that although the user knows that the solution of the problem is hidden by either user error, symptom error, or system error, she/he has no way of telling which one it is. The DETECTIVE system rests on the assumption that a systematic search through the three conditions is more productive than the other alternative, unsystematic trial and error search. The DETECTIVE system interacts with the user in quite different ways in each of the three conditions. Accordingly, we discuss each condition separately:

1. User error. There are two places where a user error is likely: (a) a mistake in man-machine interface, such as a format error in calling up the program or a data entry keypunching error or (b) a procedural error in performing a verification test. These errors may be identified by inspection and by repeating the procedures. Mundane as these errors are, everyone with programming experience knows how easy it is to overlook continually the most elementary mistake.

2. Symptom error. There is only one type of symptom error: A symptom was entered into the ISS that should not be there (i.e., we have a false symptom). The reverse -- not entering a symptom that was, in fact, present but which was overlooked -- is not a source of error. To see why this is true, consider how the interactive system works. Suppose that in creating the ISS to describe a particular equipment fault, the user identifies symptoms A, B and C, but overlooks symptom D. In generating the knowledge matrix, the system will call up those causes from its data base that share the symptom set A, B, and C. Omitting symptom D does not eliminate any possible cause from the knowledge matrix. The only harm that has been done is that the user must go through possible causes which would have otherwise been eliminated from the knowledge matrix if the system had searched for causes that share four symptoms rather than just the three.

Suppose, however, that symptom C in this same example is a false symptom; it has nothing to do with the cause of the equipment failure. For example, suppose C was a false signal created by a malfunction in the system that monitors the performance of the equipment. When C was entered into the ISS, the system automatically excluded from the knowledge matrix any cause that did not have symptom C associated with it. Thus, a number of potential causes, including in this case the actual cause, were not entered into the knowledge matrix.

The user can check for symptom error by removing the symptoms from the ISS. If all the symptoms were removed from the ISS, the system would call up every cause in its data bank. A more practical approach would be for the user to remove the symptoms from the ISS one at a time beginning with the symptom in which the user has the least confidence. For example, if the monitoring system in the example above had a history of false signals, the user might be inclined to check symptom C before the other symptoms.

3. System error. Still, we must confront the problem that in some situations, and particularly with new systems, there will be problems no one has yet encountered. There is no experimental base; therefore no knowledge matrix has been established. We shall address this error after the following discussion.

The Interrelationship of Epistemology and Heuristic

Important in this discussion is our presupposition that an epistemology (theory, classification of knowledge) is indeed a heuristic (system of/for discovering). To demonstrate this by analysis would require more space than we have available here, but since the point is important, we digress to present briefly the argument behind such a demonstration.

Of the three major questions asked by rhetoricians throughout time --Whether something is, What kind of thing it is, How can we know it (also known as the ontological, axiological, and epistemological questions) -- the

epistemological has remained the most commonly asked. Indeed, for the purpose of training, the epistemological is crucial. That the crucial question can unite the data base and the heuristic probes in the integrated system of aiding, training, and assessing is one of the bonuses of creating such a system.

The frequency drivers (Figures 11 and 12) are the basis for an epistemological system. That is, they allow one to code and classify information into unique, retrievable categories. The categories are unique in that inputs are coded such that each maintenance action on each piece of equipment is individually addressed. The categories are retrievable in that the specific address, when cross-referenced to all the rest of the information input about the activity or the equipment, can be quickly identified and pulled from the system, complete with the richness provided by nearly instant access to all other related information (that is, related by being an activity on similar equipment, similar activity on dissimilar equipment, similar time between maintenance activities, same system with slightly different activities but with the same symptom sets, and so on).

This extraordinarily rich system becomes a heuristic by the use of generative grammatical principles and generative rhetorical principles. Generative grammatical principles, especially those of the so-called rewrite transformations, provide questioning strategies which turn the cell and/or the data into meaningful questions. For example, a cell labeled (electronically) "Range" has these questions built into it: What maintenance activity (since last maintenance activity) is this? When was the fault noted? When was it repaired? By what action was it repaired? By whom? Where? What are the relationships among this cell and the other 90 cells? (That is, what arithmetic extrapolations may be derived from these relationships in other equipments, other similar scenarios, etc?) Was this activity also conducted at the subsystem and system levels in this or other similarly configured aircraft? When? Where? And so on.

FOR ALL A (e.g. 03A), AN ADDITIONAL CODING MATRIX WITH KEY ID OF SRU COULD BE NOTED.

HENCE :

LRU A

SRU

1	𝍸
2	II
3	IIII
4	II

MATRIX DRIVES "FIX"

LRU (BY ID #)

	A R&R SRU	B CLEAN CONTACTS	C REPLACE CONNECTOR	D R&R LRU
01	𝍸 I	I	III	II
02	𝍸 𝍸	I	𝍸 𝍸 𝍸 𝍸	III
03	𝍸 𝍸 III	II	II	𝍸 𝍸
04	I	I		I
05	III 𝍸 𝍸	𝍸 III	II	III
06	𝍸 𝍸 II	𝍸 II	I	𝍸

FIGURE 11 FREQUENCY MATRIX DRIVES PROBABILITY OF "FIX"

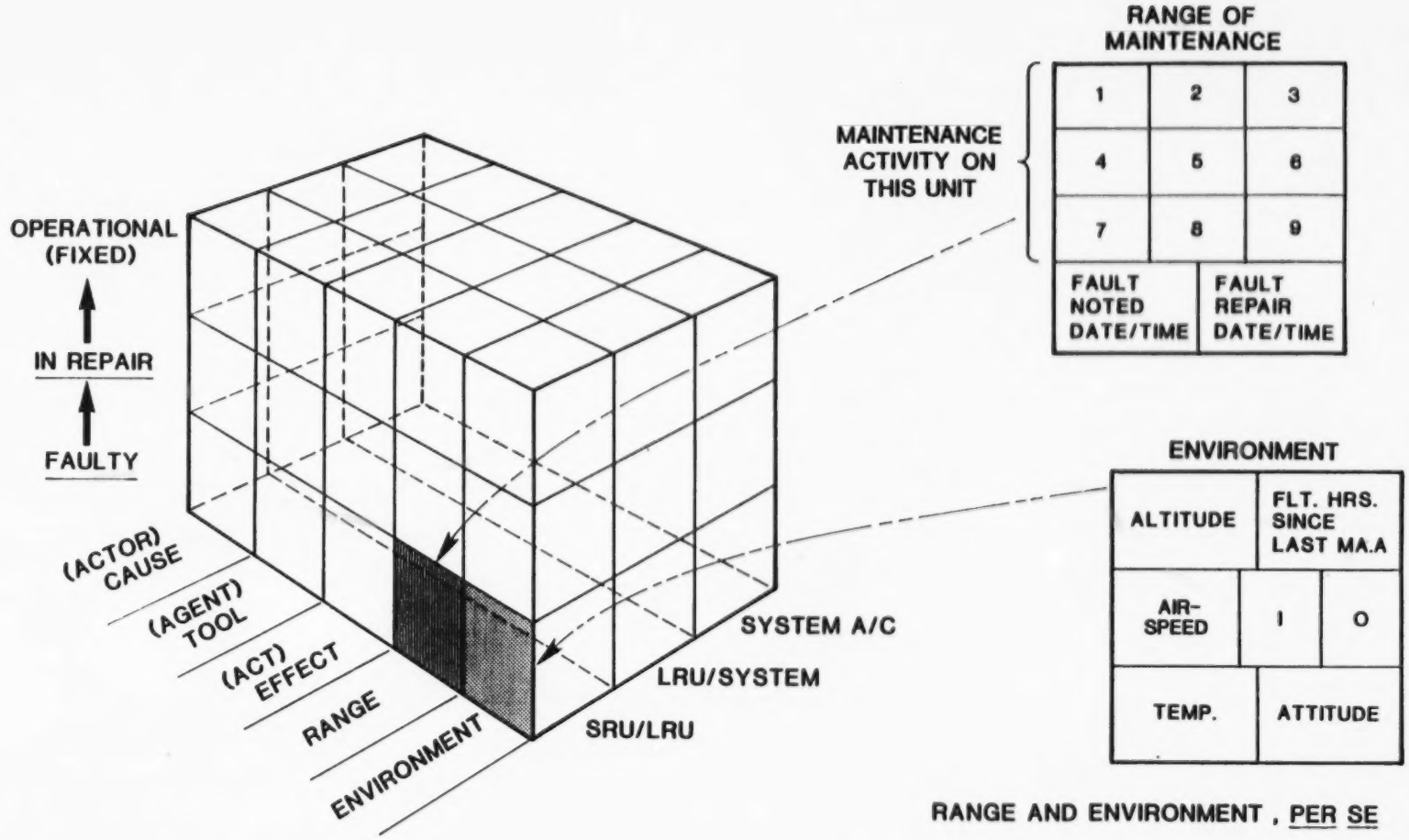


FIGURE 12 ORGANIZATIONAL SUPPORT MATRIX

Generative rhetorical principles work in the coding and subsequent retrieval (on demand) of information about the range, environment, distribution, complementary variation, definition, causes and effects, and agencies which allowed the cause to have the effect. Put more simply, the entire matrix is, in effect, a rhetorical probe.

The base of the Frequency Matrix (Figure 12) contains elements of both the Burkean and the Kintschian systems of rhetorical probes. The right side of the matrix incorporates the Young, Becker, and Pike probes, which are commonly known as the tagmemic heuristics or "tagmemic rhetoric," and which prompt the interactions which give data on complementary variation, extent, and distribution. The upper left side is a distillation of Kline's model, specifically that aspect of the model which involves questions about any activity which moves systems into or out of stasis. (For a more complete discussion of the matrix, see Kline and Huff, 1983.) The entire system is an eclectic heuristic.

The coding system is an epistemologically derived system; the model into which it is cast constitutes a rhetorically based heuristic. In that way, the proposed integrated system of aiding, training, and assessment participates simultaneously in being an epistemology and a heuristic. And, it is that intense interaction of systemic features which provides the intellectual force for the third level of DETECTIVE.

After the DETECTIVE system has verified, through authentication and field enlarging, that the probability of Type 1 (User Error) and Type 2 (Symptom) errors, respectively, has been eliminated or greatly reduced, a point is reached at which the focus of machine-man interface will, once again, shift. While in aiding mode, the machine provided data and little prompting; while in user and symptom detective error mode, the machine interacted with the human to review the decision processes of the AID mode and to elaborate the field. Now the machine becomes a prompting tool and a data recording device.

At this point, the solution is not in the machine, or -- if it be there -- it is not retrievable due to an inadequate address. In either case, the phrase "system error" is appropriate since it signals that the problem exists due to the inadequacy of the system.

After the CRT informs the operator that it is shifting into the system error mode so that others may study the problem, it begins to show sets of heuristic questions which the operator is to answer. This third level routine is designed to poll the operator for all available knowledge about the fault or problem.

The integrated system's generative grammar operates on the system's implied lexical sets to generate queries. The queries when answered are stored (both Q&A) and become part of the parameters used in fixing the probability estimates given in aiding, become new/expanded data for the system, and occasionally prompt the operator to reenter the AID or DETECTIVE sequences because the fresher picture of the problem produced refinements in the ISS or the fault sets.

A simple interrogative transformation subroutine is all that is needed. For example, if the problem has been isolated to a single LRU, our matrix is narrowed to the multicellular probe, shown in Figure 13, which becomes the lexical sets which, by action of the interrogative subroutines, yield (as an example) the questions of Figure 14. Once these questions have been answered the system will ask the operator to update user error, update symptom error, and return, if appropriate, to the ISS to rerun the program with the new information included.

If the operator does not return, the machine assigns a unique number to the case. This unique number is read into the level four training menu (after the case is reviewed by the maintenance chief); this unique number also marks the case with a direct system address, allowing the case to be found quickly.

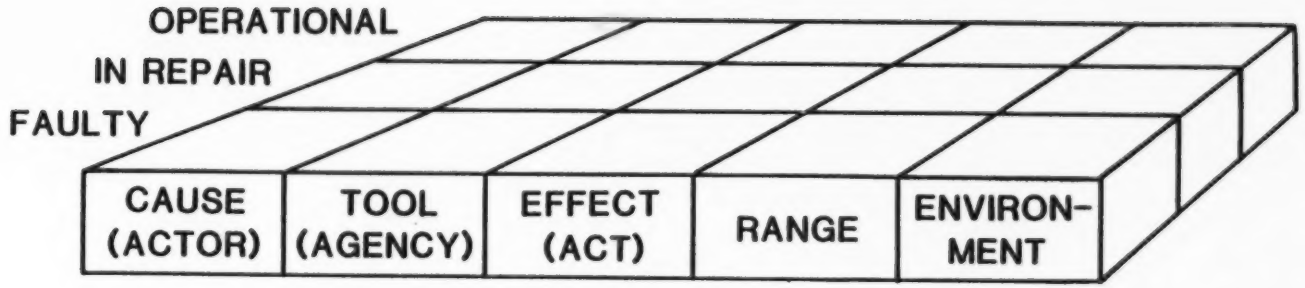
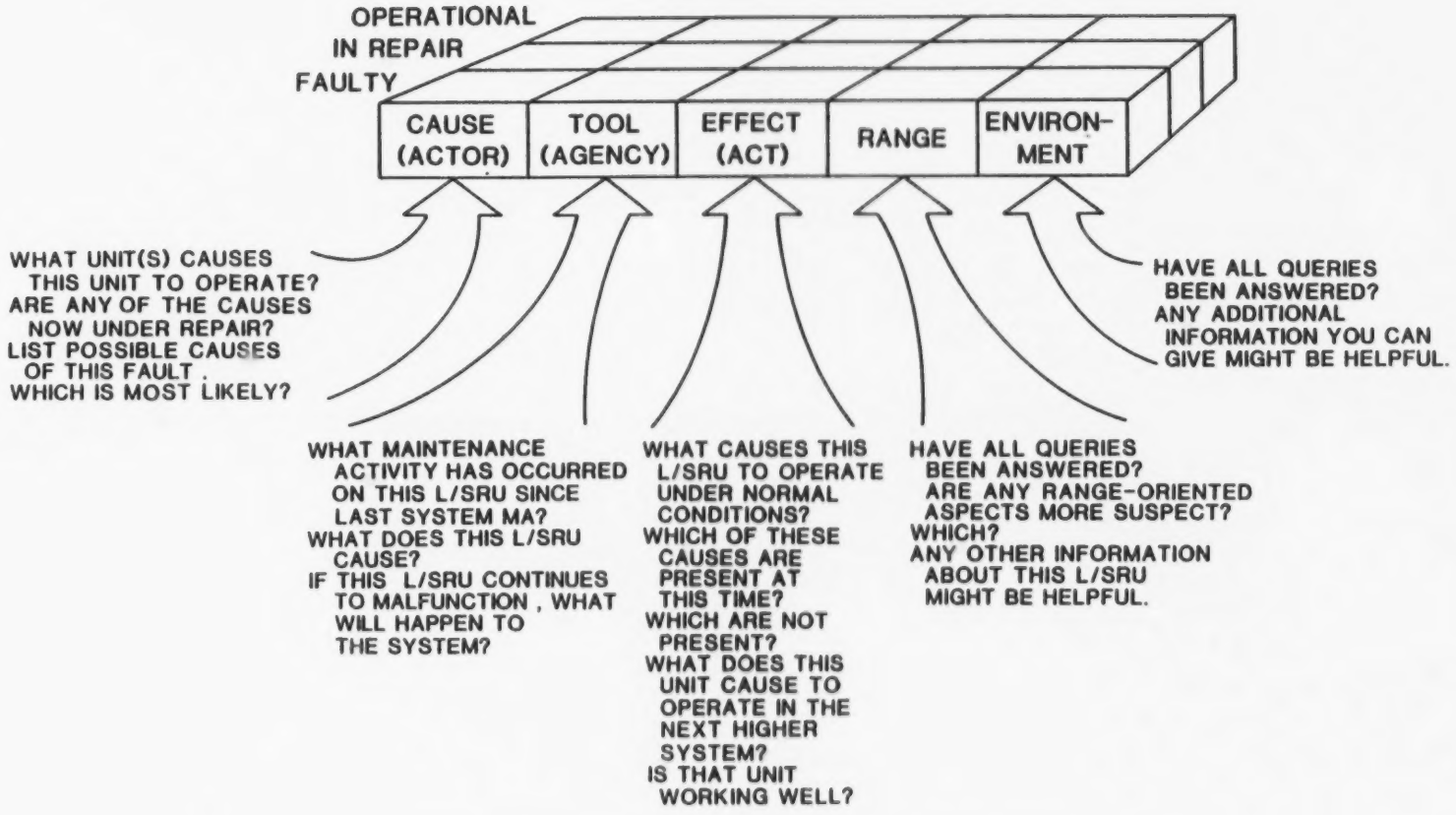


FIGURE 13 PROBE NARROWED TO SINGLE LEVEL OF CELLS


OPERATIONAL
IN REPAIR
FAULTY
CAUSE (ACTOR)
TOOL (AGENCY)
EFFECT (ACT)
RANGE
ENVIRON-MENT
WHAT UNIT(S) CAUSES THIS UNIT TO OPERATE?
ARE ANY OF THE CAUSES NOW UNDER REPAIR?
LIST POSSIBLE CAUSES OF THIS FAULT.
WHICH IS MOST LIKELY?
WHAT MAINTENANCE ACTIVITY HAS OCCURRED ON THIS L/SRU SINCE LAST SYSTEM MA?
WHAT DOES THIS L/SRU CAUSE?
IF THIS L/SRU CONTINUES TO MALFUNCTION, WHAT WILL HAPPEN TO THE SYSTEM?
WHAT CAUSES THIS L/SRU TO OPERATE UNDER NORMAL CONDITIONS?
WHICH OF THESE CAUSES ARE PRESENT AT THIS TIME?
WHICH ARE NOT PRESENT?
WHAT DOES THIS UNIT CAUSE TO OPERATE IN THE NEXT HIGHER SYSTEM?
IS THAT UNIT WORKING WELL?
HAVE ALL QUERIES BEEN ANSWERED?
ARE ANY RANGE-ORIENTED ASPECTS MORE SUSPECT?
WHICH?
ANY OTHER INFORMATION ABOUT THIS L/SRU MIGHT BE HELPFUL.
HAVE ALL QUERIES BEEN ANSWERED?
ANY ADDITIONAL INFORMATION YOU CAN GIVE MIGHT BE HELPFUL.

INTERROGATIVE ROUTINE PROVIDES, PROMPTS AND QUERIES IN THE DETECTIVE MODE.

FIGURE 14 INTERROGATIVE ROUTINE PROVIDES PROMPTS AND QUERIES IN THE DETECTION MODE.

BEST COPY AVAILABLE

Finally, the operator is instructed to log off.

Whenever a solution is found, the case is written up completely and sent back through the system to the address. In this way, the newly discovered solution is quickly integrated. The unique case number is erased when the solution is encoded; this takes the case out of the level four training menu and leaves the case in the data bank.

In these ways, job aiding with continuously updating data bases is possible. It is these updated data bases which provide the rich source of training materials, the features of which are shown in Figure 15.

B. Training

Training is crucial to any organization, and certainly the Air Force is no exception. The training program must provide realistic training, thus eliminating the "Transfer of Learning" problem. It must provide up-to-date instruction which uses all current Technical Orders (TOs) and Training/ Technical Manuals; thus it must be rapidly and directly updatable. It must provide trainees with opportunities for additional, personally chosen training; thus, it must be both user-friendly and accessible. Our conception of the Integrated Aiding, Training, and Assessing Model provides these features.

The training system is drawn from the military training model. Instructional development is a twelve step process in our view.

1. Perform task/job analysis.
2. Draft preliminary objectives (Performance objectives; who does what, with what, to what level?).
3. Set and confirm prerequisites for instruction.
4. Determine students' background (in general).
5. Revise objectives.
6. Design test procedures.
7. Select materials.

TRAINING

- REALISTIC, HANDS-ON
- UP-TO-DATE
- USER-FRIENDLY
- ACCESSIBLE

FIGURE 15 FOUR FEATURES OF TRAINING

8. Sequence materials.
9. Select instructional method(s).
10. Teach.
11. Test.
12. Revise as necessary to achieve program goals (100/100%, 90/100%*).

Instructional development, additionally, must fit into the Integrated Logistics Support (ILS) system. Within the ILS Model, there are provisions for three major influences/inputs into training: Instructor, ILS data (which include Technical and Training Manuals and Reliability and Maintainability Ojectives), and training course design (usually provided by contractors). Thus, the Air Force and the contractor cooperate on the first five steps; the contractor provides the next three; and the Air Force the four following. The integrated model is focused on the AF delivery portion (the last four steps).

The instructional methods we incorporate are direct teaching (via manuals and other printed materials and instructor presentation), case study, and problem solving. We shall omit discussion of the instructor in the balance of this paper since the system is designed to provide meaningful On-the-Job Training (OJT), which supports an instructor if one is available.

How the Training Subsystem Works

When a trainee keys into the CRT -- whether done because an instructor or superior required the work or because the trainee wants to know about a unit or system upon which work may someday be required -- he or she is presented a menu which lists the available materials about the systems and subsystems. After keying to the (sub) system, the trainee finds a second menu; this one has an option for training and one for job aiding (plus others which may later be added). The training option is selected.

---

*100% of trainees score 100% correct, 90% of trainees score 100% correct.

The CRT then displays a message something like this:

A1/C SMITH:
FOR TRAINING YOU HAVE THE OPTION OF GOING DIRECTLY TO THE CASE STUDIES AND ANALYSES OF MAINTENANCE ON THE _____ LRU/SYSTEM OR USING THE LIBRARY TO READ AND STUDY ABOUT LRU/SYSTEM. DO YOU WISH TO USE THE LIBRARY? Y/N

The library has indexed options for all the TMs, TOs, Ilustrated Parts Breakdowns, charts, graphs, and schematics -- along with recommended reading lists for introductory, advanced, and current awareness reading. The library also has a sample listing of job aids available on the indexed subject. The trainee may read the materials on the CRT or by securing a hard copy (on a remote duty station); for current awareness materials, the system may be instructed to print any material. After library use or instead of using it, the trainee can move on to the case study problems.

The case studies are drawn from the job aids. In the training mode, both the solution and the algorithm are being taught. Each of the steps in the job aid is available in turn -- when the trainee requests them. These options can be presented after each step is completed.

In the training mode case studies are arranged by level of difficulty (see Figure 16) into four sets:

Introductory. In these, the solution is the option which has high probability and cost-effectiveness.

Intermediate. In these, the solution is discovered to be a lower probability (with or without cost trade-off).

Advanced. In these, the solution is found during the heuristic search, and it is of mid to high probability.

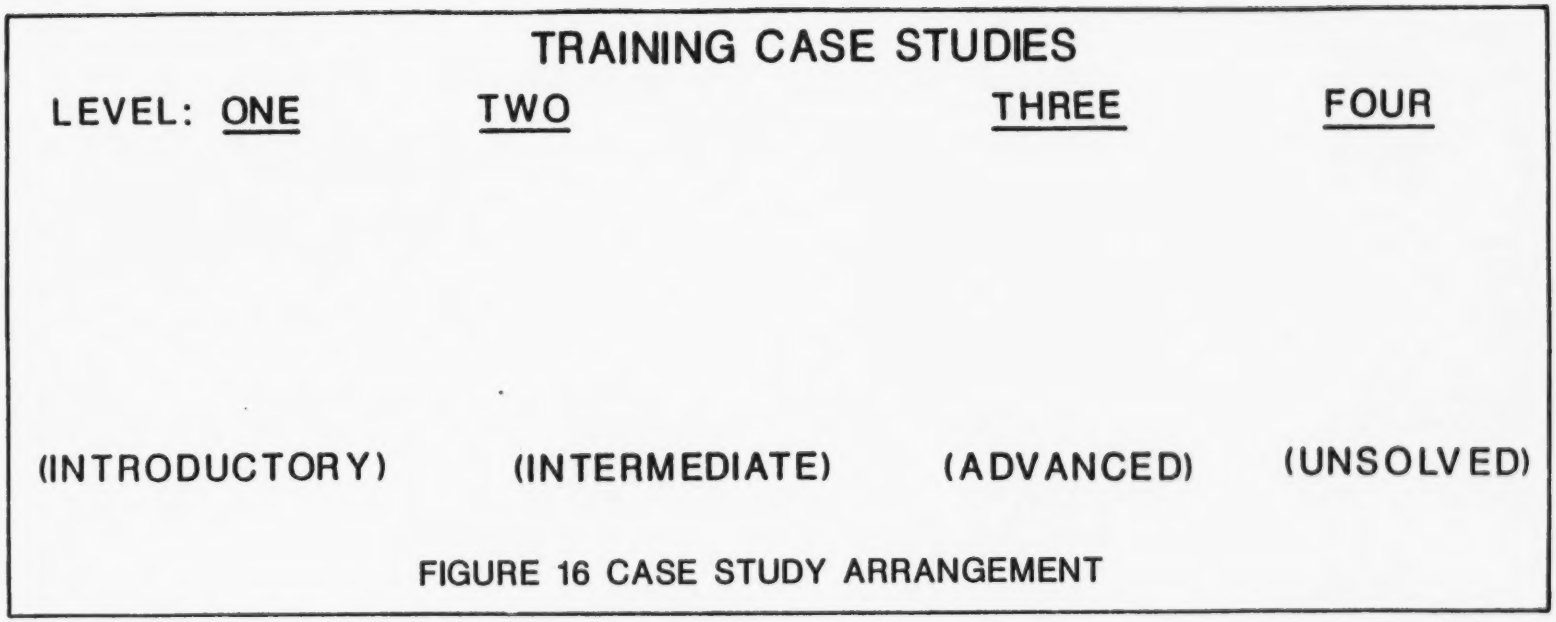


FIGURE 16 CASE STUDY ARRANGEMENT

Yet Unsolved. An actual, unsolved problem currently under study.

While solving a problem in a case study the trainee will have all standard job aid tools available (and, perhaps, an instructor).

Trainees can produce a hard copy of their completed case study (to present to their instructor or the maintenance chief) or of the case per se. For level four cases, the Air Force may wish to offer a cash prize for a defensible solution; this would be to encourage discussion of the problem -- especially discussion by veteran flight-line non-commissioned officers (NCOs).

In addition to the case studies, which are a form of testing, the trainee can call up self-tests from a menu. These are drawn from training manuals, SOPs, and the case studies and are the tests which may be used in resident instruction classes.

They serve as performance indicators for "passing out" of a training requirement; they also allow continuing availability of an array of tests on basic avionics circuitry, logic and microwave circuit fundamentals, electronic warfare/command-control-communication interfaces, fuels, and other issues of concern. These tests can be used by NCOs and officers as screening tools for replacements, by anyone for self-testing (before beginning a course, for example), and by instructors as a source of daily tests and pre-/post-tests.

When case studies and supplementary tests are packaged separately, they become the battery for performance assessment.

C. Performance Assessment

Interrelationship of Teaching and Testing

An especially important presupposition in the model is the direct interrelation of teaching and testing: Any procedure, protocol, or exercise

which can be used as a teaching device can also be used as a testing tool.

For example: Teach $2 + 2 = 4$; $2 + 3 = 5$.
Test $2 + 2 = \underline{?}$; $\underline{?} + 3 = 5$.

Or: TEACH The characteristic impedance of a parallel transmission line may be determined by solving $Z_{\emptyset} = 276 \log \frac{b}{a}$, where b is the distance between conductors and a is the diameter of one of the conductors.

TEST What is the characteristic impedance of a parallel transmission line with four inch spacers and .023 drawn copper wire conductors?

This presupposition is important because the elements of the training exercises become part of the battery offered for performance assessment. The machine has no problem with this translation, since the conversion rules are those of transformational grammar, notably the "T-WH," "T-DO," and "T-BE, Present" rewrite rules.

Performance assessment involves many considerations, ranging from aptitude, general knowledge, to time in service/grade/role, in order to actually demonstrate performance. The case studies and the array of tests available in the training mode, added to other, already existing performance assessment tools provide an enriched and constantly updatable source of tools for assessing performance -- both actually demonstrated and potential performance.

For routine screening, the introductory case studies from training would be more appropriate as frames for determining whether the troubleshooting algorithm has been internalized. The actual training case

study may be used to authenticate performance capabilities. And advanced or yet unsolved cases may be used as exit criteria for noting whether higher level (training or AFSC) assignment is warranted.

We believe it is important for individuals to have access to performance assessment batteries, i.e., self-assessment. This is an especially important consideration when it can be combined with a self-directed training program. In this integrated model that is possible.

Richardson (1983) pointed out:

> The challenges are to move performance measurement into the operational environments and to integrate it with personnel management and training organizations' programs. The development of symbolic substitutes, based on modern maintenance simulation work (seen as interactive graphics simulation utilizing intelligent video discs), also provides a challenge. (p. 35)

The proposed integrated model, in fact, does move this role into the operational environment.

Special attention is due for Richardson's second "challenge." The routine use of the (anticipated) software-based simulations of virtually all test, maintenance and troubleshooting systems/procedures will be focused particularly on performance assessment roles. With the updatable, integrated system, we anticipate that the use of simulations will be optimized.

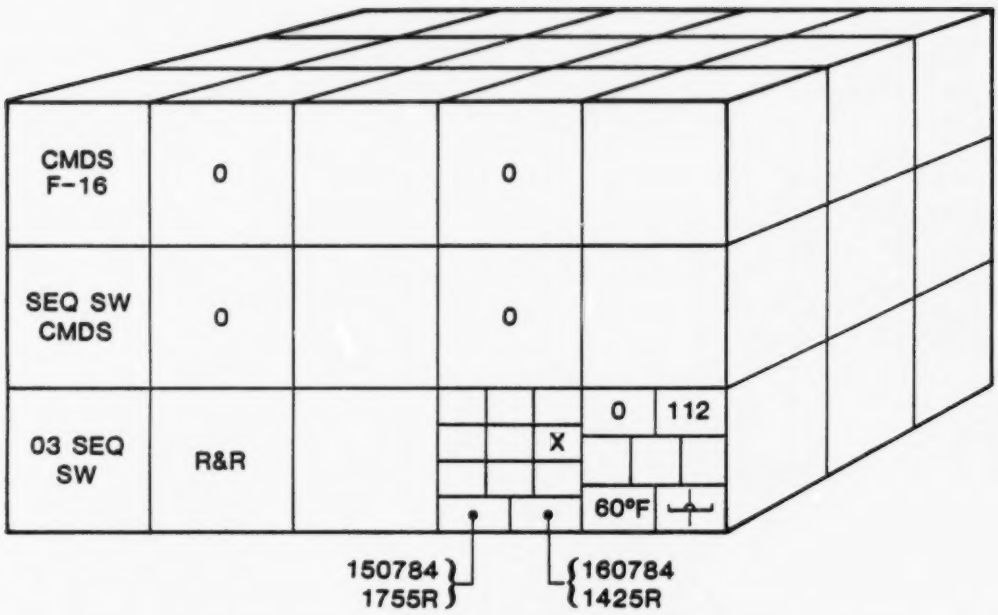
III. UPDATING THE SYSTEM

Although it is somewhat beyond the scope of this paper to address the infrastructure of the tool used to update this integrated system, it is nonetheless important to dwell briefly on this problem.

We provide conceptually designed frameworks for updating LRU/SRU fix probability (see Figure 11), for Mean Time Between Failures (MTBF) and case histories (see Figure 12), for Built-in Test (BIT fault isolation tracing (see Figure 17), and for system scanning data (see Figure 18). The lowest level is the frequency count which drives the fix probability table (using simple r statistics). Each datum has an unique address; so from the histogrammic data in the fix table an individualized SRU and LRU can be traced, even while maintenance data are being aggregated. The Fault Isolation Tree is driven by BITE, by system/subsystem flow diagrams, by TOs, and by user provided information. Organizational support data are provided within the matrix. From the matrices of many maintenance activities can be drawn aircraft system, LRU, and SRU data, similar cause/similar effect/similar context data, as well as multiple test and repair data.

Finally, system scanning Beta (the system studying itself, that is) ensures routine Macro-MTBF data collection.

Updating can be by direct input, by teletype, by batch load -- all that is required is state-of-the-art software and prearranged availability (for example, monthly).



(LEGEND : SRU 03 OF THE SEQUENCER SWITCH , COUNTERMEASURES DISPENSING SYSTEM (ON AN F-16), WAS REMOVED AND REPLACED. THE MALFUNCTION WAS NOTED ON JULY 15, 1984 AT 5:55 PM ; AT THE TIME OF THE ACCIDENTAL FIRING, THE F-16 WAS ON THE GROUND (0 ALTITUDE, STABLE, STATIONARY ATTITUDE) AND THE AMBIENT TEMPERATURE WAS 60°F. THE LAST MAINTENANCE ACTIVITY ON THIS LRU WAS 112 FLIGHT HOURS AGO; THERE HAVE BEEN SIX MAINTENANCE ACTIVITIES ON THIS LRU (THE X IN SQUARE 6 OF THE "RANGE" BLOCK). THE SYSTEM WAS REPAIRED AND INSPECTED AT 2:25 PM ON JULY 16,1984).

FIGURE 17 MATRIX TRACE OF SAMPLE SRU R&R

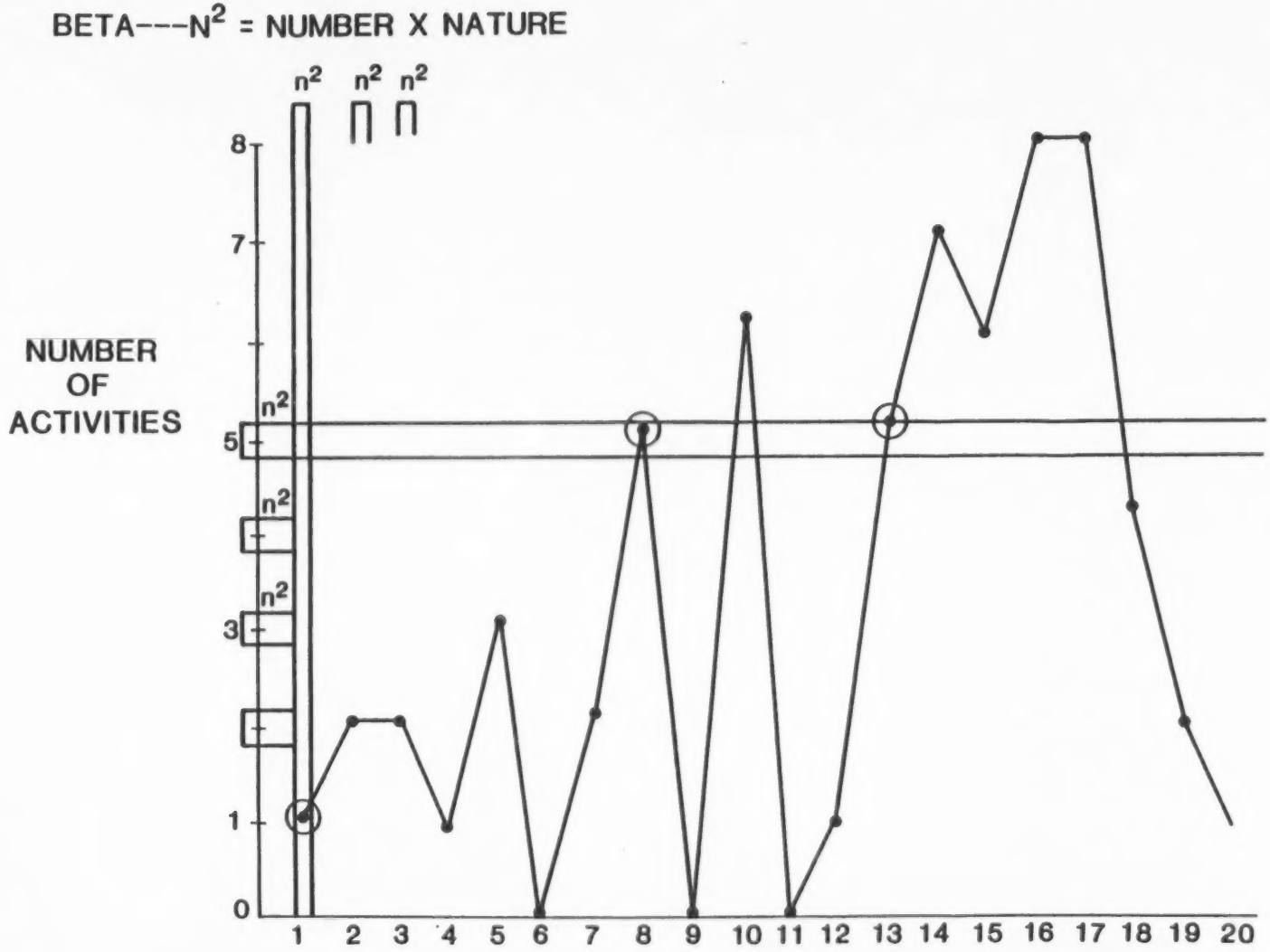


FIGURE 18 MAINTENANCE ON LRU (BY ID NO.)

IV. ANTICIPATING CLEARLY THE FUTURE

The program envisioned in this paper is not a solution to a problem which exists now and once solved will go away. Indeed, the problem will continue and will change. The solution, then, must be one which can change and can still provide responses in meaningful formats. While we are uncertain of the future and problems yet to become known, we have considered eight specific innovations/new problems: Four will be confronted we believe before 1995, and four will begin to be known and be confronted around and after 1995.

A. Present - 1995

1. "Smarter" ATE/STE. Systems already under development and due for production in late 1988 and 1989 will have interactive software. This is especially true for specialized test equipment (STE) and somewhat true for automatic test equipment (ATE). For example, a design proposed for the ALE-47 countermeasures dispensing set will require only one piece of STE (and that is of a complexity currently within reach); it will require ATE currently in production with a new software package -- this software will enable more testing, quicker fault isolation, and compatibility with the ALE-47 software, which is extraordinarily complex.

The E/EF/F-111 aircraft, for another example, will use in its EW suite some six major busses (not counting edge-light and main source busses). ATE used for fault isolation will "read" all six busses simultaneously (necessarily) and interactively. STE must, of course, do likewise.

Smarter ATE and STE will draw heavily on AI breakthroughs currently being reported and tested. Systematic search routines, quicker because they are not serial or linear, are already being incorporated.

The proposed system will work well with the new ATE/STE and will have compatible data inputs/outputs (I/Os). The compatibility is assured because the matrix allows interactive collection of information. Only the address must be coded.

2. "Natural Language" I/O. Between the present and 1995, we predict that the current research in natural language I/Os will produce key word identification systems more nearly representative of a true, "natural" language system. The principal benefit of this is user-friendliness. (User-friendliness increases as the nature and the quantity of encryption required decrease.)

While these systems and their related syntactic parsers will move us much closer to natural language I/Os, actually achieving that state remains remote -- at least within the next ten years.

3. Expanding Data Base. The problem most difficult to imagine is the data base in 1993 to 1995 (and beyond). Current Air Force programs will still be invaluable in that data base, but the newer systems will outnumber the existing ones. These new systems include the small ICBM, the revised MX, the B-1B, Stealth, the U-4, integrated EW/RWR/WS and MAD (Missile Approach Detector Systems), PenAids, the "new" UYK-19, Advanced MIL-STD-1750A CPU architecture -- all these and more!

Providing for this data base is mind-boggling in terms of quantity but not concept. The proposed system will not be burdened by the new data systems. The major problem will be formatting inputs -- and this should be levied against system contractors on the Contract Data Requirements Lists (CDRLs). Updating will then, of course, be required.

From 1995 to 2005 the system load will again shift as the Anti-Satellite Weapons systems are completely replaced; this replacement, however, will not disrupt the proposed system.

4. Job Aiding, Training, and Simulators. No doubt, simulators will become the most commonplace delivery vehicle for all forms of training. The simple formula of available use + cost + learning dictates this clearly.

Since this system is designed to be based on job aiding and training, as newer simulators and simulations arise there will be no problems in adapting to them.

5. Integrated Expert Systems and Local Area Networks (LANs). As the WWMCCS Information System (WIS) and the ULANA, AWIS, and NAVLAN (for, respectively, the Air Force, the Army, and the Navy) come on line in 1986-1992, the system proposed in this document will become fully deployable. The distributed intelligence characteristics of the WIS Blocks D through N suits the proposed system well.

The interface device to be used in the Air Force architecture, and likely for the WWMCCS Information System Network, will probably not be application-specific. Hopefully its procurement (probably amounting to some 50,000 units in the years up to 1992 and the same number in the years from 1992 to 2000) will be based on its versatility. The $.5B (or so) spent in the next seven years on those interface devices will, in fact, be the implementing funds for a system such as the one proposed in this document.

6. Test Bed. In order to prepare for the applications envisioned and to be ready for Blocks C and D of the WWMCCS Information System, where the real testing of both the Information System and the proposed model will be possible, we strongly recommend that the Air Force and AFHRL continue to support such programs and move them to test-bed phases.

B. Beyond 1995

In addition to the developments noted above, we anticipate at least four others to become critical in the decade or so beginning around 1995.

1. Ops BITE Download. Currently planned BITE improvements include moving data with BITE and "BITE-interfaced off-loading" from aircraft system to the test bench. Long a favorite idea of dreamers and science fiction writers, BITE download is now within reach. In this plan, the pilot, engineering officer, and/or the crew chief will transmit on-board BITE to the test bench, as they depart the A/C. (Not only A/C need be considered; any system can be involved.) Test bench sets can poll BITE while the system is in use. (One interesting concept is the continuously down-linked telemetric BITE -- stored for extraordinarily high-speed burst transmissions over encrypted carriers in tactical situations.)

The system proposed in this paper is compatible for the same reason it is in A-3, above. Formatting I/Os will be very time consuming and expensive. These will be levied on contractors as part of R&M Program Plans and of Software Specifications.

2. Universal ICAI. "Universal" or, perhaps, "nearly universal" Intelligent Computer-Assisted Instruction (ICAI) will be the case in the years beyond 1995. This use of intelligent systems is not a problem for the plan advocated in this paper. Rather, it is the base toward which this plan moves.

3. Hemispheric Hook-ups. By 2005, individual test benches probably will be part of hemispheric or worldwide networks. The increasingly short turnaround pooling will be an asset to the integrated aiding, training, assessing model.

Since electronic intercommunication does not present any technically insurmountable problems now, we anticipate little problem in implementing this "intercontinental test bench."

4. Hand-Held/Helmet-Mounted Units. These are not real technical challenges; they are "innovative" in the sense that they allow (and probably promote) remote accessing of flight-line gear. Problems in projection at varying light conditions are the problem presently and will continue to be. No problems are presented for the proposed system, however.

V. CONCLUSION

The programming proposed in this paper requires further study, especially in the areas of life cycle costs, initial funding and dissemination, and software development. The model itself should be studied from four perspectives:

- Routines and subroutines required
- Matrix - address algorithm
- Scope of library requirements
- Plan for further study

While budgeting and forecasting using current fiscal year dollars is virtually impossible, we assert that, including initialization, the Air Force (alone) can expect that savings directly attributable to this program will exceed $100 million per annum. Adding the Army and Navy to the Air Force in a Tri-Service Initiative would create extraordinary savings.

REFERENCES

Chomsky, N. (1963). Formal properties of grammars. In R. Luce, R. Bush, and E. Galantner (Eds), Handbook of Mathematical Psychology: Vol. II. New York: John Wiley and Sons.

Kline, C.R., & Huff, R.K. (1983). Reading, writing: Radix. Visible Language, XVII, 2, 163-176.

Richardson, J.J. (1983, September). Artificial intelligence: An analysis of potential applications to training, performance measurements, and job performance aiding (AFHRL-TP-83-28, AD-A133592). Lowry Air Force Base, CO: Air Force Human Resources Laboratory, Training Systems Division.

SELECTED BIBLIOGRAPHY

Anderson, J.R. (1982). Acquisition of cognitive skill. Psychological (Review, 89), 369-406.

Antonelli, D.R. (1984). The application of artificial intelligence to a maintenance and diagnostic information system. Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL TR-84-25, AD-A145 349). Lowry AFB, Colorado: Air Force Human Resources Laboratory, Training Systems Division.

Backus, J. (1978). Can programming be liberated from the Von Neumann style? A functional style and its algebra of programs. Communications of the Assoc. for Comput. Machinery, 21, 8, 613-641.

Bailes, P.A., & Reeker, L.H. (1980a). Post-X: An experiment in language design for string processing. Australian Computer Science Communications, 2, 2, 252-267.

Bailes, P.A., & Reeker, L.H. (1980b). An experimental applicative programming language for linguistics and string processing, Proceedings, 8th International Conference on Computational Linguistics (pp. 520-525). Tokyo, Japan.

Baker, M., & Hamovitch, M. (1983). Job-oriented basic skills (JOBS) training program: An evaluation (NPRDC Tech. Rep. 83-5). San Diego, CA: Navy Personnel Research and Development Center.

Brown, J.S. (1983). Learning by doing revisited for electronic learning environments. In M.A. White (Ed.), The future of electronic learning. Hillsdale, NJ: Erlbaum.

Bylander, T., Mittal, S., & Chandrasekaran, B. (1983). CSRL: A language for expert systems for diagnosis. IJCAI-83.

Carbonnel, J. (1970). AI in CAI: An artifical intelligence approach to computer-assisted instruction. IEEE Transactions on Man-Machine Systems, MMS-11(4).

Chandrasekaran, B., Gomez, F., Mittal, S., & Smith, J.W. (1979). An approach to medical diagnosis based on conceptual structures, IJCAI-79.

Chase, W.G., & Ericsson, K.A. (1982). Skill and working memory. In G. Bower (Ed.), The psychology of learning and motivation (Vol. 16). New York: Academic Press.

Clancey, W.J. (1981). The epistemology of rule based expert system: A framework for explanation (Report #CS-81-896). Stanford, CA: Dept. of Computer Science, Stanford University.

Clocksin, W.F., & Mellish, C.S. (1981). Programming in PROLOG. Berlin: Springer-Verlag.

de Kleer, J., & Brown, J.S. (1981). Mental models of physical mechanisms and their acquisition. In J. R. Anderson (Ed.), Cognitive skills and their acquisition. Hillsdale, NJ: Erlbaum.

de Kleer, J., & Brown, J.S. (1983). Assumptions and ambiguities in mechanistic mental models. In D. Gentner & A. L. Stevens (Eds.), Mental models. Hillsdale, NJ: Erlbaum.

Dallman, Brian (1984). AFHRL program for artificial intelligence applications to maintenance and training. Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-A145 349). Lowry AFB, Colorado: Air Force Human Resources Laboratory, Training Systems Division.

Duffy, T.M., & Kabance, P. (1982). Testing the readable writing approach to text revision (NPRDC Technical Report). San Diego, CA: Naval Personnel Research and Development Center.

Duffy, T.M., Sass, D., & Curran, T.E. (1983). Job reading tasks and the design of technical text. Paper presented at the meeting of the American Education Research Association, New York.

Feigenbaum, E.A., & McCorduck, P. (1983). The fifth generation. Reading, Massachusetts: Addison-Wesley.

Gentner, D. (1983). Structure mapping: A theoretical framework for analogy. Cognitive Science, 7, 155-170.

Greeno, J.G. (1978). A study of problem solving. In R. Glaser (Ed.), Advances in instructional psychology (Vol. I). Hillsdale, NJ: Erlbaum.

Greeno, J.G. (1980). Some examples of cognitive task analysis with instructional implications. In R.E. Snow, P.A. Federico, & W.E. Montague (Eds.), Aptitude, learning and instruction: Vol. 2, Cognitive process analyses of learning and instruction. Hillsdale, NJ: Erlbaum.

Griffin, A.F. (1984). Intelligent information retrieval from on-line technical documentation. Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-145 349). Lowry AFB, Colorado: Air Force Human Resources Laboratory, Training Systems Division.

Griswold, R.E. (1984). The control of searching and backtracking in string pattern matching. In Campbell (Ed.) Implementations of PROLOG (pp. 50-64). Chichester, England: Ellis Horwood.

Harrison, M.A. (1978). Introduction to Formal Language Theory. Reading, Massachusetts: Addison-Wesley.

Honeywell, Inc. (1983). Reference Manual for the Ada Programming Language, ANSI/MIL-STD-1815A. Produced for the United States Department of Defense by Honeywell Systems and Research Center, Minneapolis, and Alsys, La Celle Saint Cloud, France.

Honeywell, Inc. (1984). Rationale for the Design of the Ada Programming Language. Produced for the United States Department of Defense by Honeywell Systems and Research Center, Minneapolis, and Alsys, La Celle Saint Cloud, France.

Johnson, R.C. (1981, September). Integrated maintenance information system: An imaginary preview (AFHRL-TP-81-18, AD-104 224). Wright-Patterson AFB, OH: Air Force Human Resources Laboratory, Logistics and Technical Training Division.

Kay, M. (1980). Algorithm schemata and data structures in syntactic processing (Technical Report). Palo Alto, CA: XEROX Palo Alto Research Center.

Lewis, H.R., & Papadimitriou, C.H. (1981). Elements of the theory of computation. Englewood Cliffs, New Jersey: Prentice-Hall.

Markov, A.A. (1951). Theory of algorithms. Trudy Mathematicheskogo Instituta imeni V. A. Steklova, 38, 178-189. English Translation in American Math. Society Trans., 2, 15, 1-14 (1960).

Pereira, F.C.N., & Warren, D.H.D. (1983). Parsing as deduction. Proceedings of the 21st Annual Meeting of the Association for Computational Linguistics. Cambridge, MA.

Psotka, J. (1984). Artificial intelligence contributions to training and maintenance. Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-145 349). Lowry AFB, Colorado: Air Force Human Resources Laboratory, Training Systems Division.

Reeker, L.H., Kreuter, J., & Wauchope, K. (1985). Artificial intelligence in ADA: Pattern-directed processing (AFHRL-TR-85-12, AD-156 230). Lowry AFB, Colorado: Air Force Human Resources Laboratory, Training System Division.

Reinstein, H.C., & Aikins, J.S. (1981). Application design: Issues in expert system architecture. Proceedings, 7th International Joint Conference on Artificial Intelligence (pp. 888-892). Vancouver, Canada.

Richardson, J. Jeffrey. (1983, September). Artificial intelligence: An analysis of potential applications to training, performance measurement and job performance aiding (AFHRL-TP-83-28, AD-133 592). Lowry AFB, Colorado: Air Force Human Resources Laboratory, Training Systems Division.

Rieger, C., & Small, S. (1979). Word expert parsing. Proceedings, Sixth International Conference on Artificial Intelligence. Tokyo, Japan.

Rogers, Hartley, Jr. (1967). The theory of recursive functions and effective computability. New York, NY: McGraw-Hill.

Rumelhart, D.E., & Ortony, A. (1977). The representation of knowledge in memory. In R.C. Anderson, R.H. Spiro, & E. Montague (Eds.), Schooling and the acquisition of knowledge. Hillsdale, NJ: Erlbaum.

Shank, Roger C. (1982). Dynamic memory: A theory of learning in computers and people. Cambridge: Cambridge University Press.

Smith, E.E., & Goodman, L. (1982). Understanding instructions: The role of explanatory material (Technical Report No. 5088). Cambridge, MA: Bolt Beranek and Newman, Inc.

Szolovits, P., Patil, R., & Schwartz, W. (1981). Causal understanding of patient illness in medical diagnosis. IJCAI-81 (August), 893-899.

Valiant, L.G. (1975). General context-free recognition in less than cubic time. Journal of Computer and Systems Sciences, 10, 308-315.

Van Baalen, J. (1984). On the Requirements of Expert Systems of Fault Isolation, Artificial Intelligence in Maintenance: Proceedings of the Joint Services Workshop (AFHRL-TR-84-25, AD-145 349). Lowry AFB, Colorado: Air Force Human Resources Laboratory Training Systems Division.

VanLehn, K. (1983). Felicity conditions for human skill acquisition: Validating an AI-based theory (Tech. Rep. CIS-21). Palo Alto: Xerox Palo Alto Research Center.

Wescourt, K.T., Beard, M., Gould, L., & Barr, B. (1977). Knowledge-based CAI; CINS for individualized curriculum sequencing (Tech. Rep. No. 290). Stanford, CA: Stanford University, Institute for Mathematical Studies in the Social Sciences.

Winston, P.H., & Horn, B.K.P. (1981). LISP. Reading, Massachusetts: Addison-Wesley.

Woods, W.A. (1970). Transition network grammars for natural language analysis. Communications ACM., 13.

Yngve, Victor H. (1958). A programming language for mechanical translation. Mechanical Translation, 5, 1, 25-41.